Collocating Local Volatility Model

K. Spanderen
Uniper Global Commodities

QuantLib User Meeting 2016
Düsseldorf
08.12.2016

QuantLib Integration I: Multi-Threading

- The pricing library is mainly JVM based with QuantLib being integrated via SWIG
- Hosted in WebContainers on multiple NUMA machines

  ⇒ Thread-safe code is crucial
- Communication: JSON or XML via http or WebSockets
- Do not share QuantLib objects between threads

  ⇒ use Riccardo’s thread local singleton pattern
- JVM has multi-threaded garbage collector

  ⇒ set QL_ENABLE_THREAD_SAFE_OBSERVER_PATTERN

QuantLib Integration II: Memory Performance

- Important use case: Large number of parallel pricing via PDE
- Observation on large NUMA Windows server:

  Kernel consumes almost all CPU time
- Many new/delete calls with same array size

  ⇒ Malloc does not scale
- Solution: thread local "cache" for Array and Matrix

  ⇒ released memory is kept in thread local list
  ⇒ next Array related malloc reuses thread local memory

QuantLib Integration III: Garbage Collection

Memory Bombshell

```
for (i <- 0 until 1000000)
  new org.quantlib.Array(125000)
```

QuantLib Integration III: Garbage Collection

- Code sniplet is likely to shutdown your JVM
- Memory allocated by QuantLib Array is unknown to JVM
- JVM will not trigger garbage collector even if the OS is already running out of memory
- Workaround from SWIG manual does not work on recent JVMs

Solution from the ’90s

```
for (i <- 0 until 10000000)
  (new org.quantlib.Array(125000)) delete
```

Introduction: Different View on Calibration

Risk neutral pricing with implied probability density function $f(S_t, t)$

$$V_{call}(t, K) = e^{-rT} \int_K^{\infty} (S_t - K) f(S_t, t) dS_t$$

By differentiation of the equation above, the market implied cumulative distribution function $F_{S_t}(s)$ is given by

$$\begin{aligned} F_{S_t}(s) &= Prop(S_t \leq s) = \int_0^s f(S_t, t) dS_t \\ &= 1 + e^{rt} \frac{\partial V_{call}(t, K)}{\partial K} \mid_{K=s} = \; e^{rt} \frac{\partial V_{put}(t, K)}{\partial K} \mid_{K=s} \end{aligned}$$

Model calibration: Tune model parameters to match $F_{S_t}(s)$.

Exact Calibration: Local Volatlility Model

In the Local Volatility (LV) model the volatility $\sigma_{LV}(S, t)$ is a function of spot level S_t and time t. The dynamics of the spot price is given by:

$$dS_t = (r_t - q_t)\, S_t dt + \sigma_{LV}(S_t, t) S_t dW_t$$

$$\sigma_{LV}^2(S, t) = \left. \frac{\frac{\partial C}{\partial T} + (r_t - q_t)\, K \frac{\partial C}{\partial K} + q_t C}{\frac{K^2}{2} \frac{\partial^2 C}{\partial K^2}} \right|_{K=S, T=t}$$

The model is often criticized for its unrealistic volatility dynamics.

The Dupire formula is mathematically appealing but also unstable.

Lech Grzelak: Collocating Local VolatlilityModel

The dynamics of the spot process is given by some kernel process X_t and a deterministic mapping function $g(X, t)$ such that

$$\begin{aligned} S_t &= g(t, X_t) \\ dX_t &= \mu(X_t, t)dt + \sigma(X_t, t)dW_t \end{aligned}$$

The mapping function $g(t, x)$ is chosen such that the cumulative distribution function (CDF) of S_t matches the market implied CDF.

The LV model is a special case of the CLV model with

$$\begin{aligned} g(t, X_t) &\equiv X_t \\ \mu(X_t, t) &= (r_t - q_t)\, X_t \\ \sigma(X_t, t) &= \sigma_{LV}(X_t, t)X_t \end{aligned}$$

Collocating Local Volatlility (CLV) Model

- The choice of the stochastic kernel process X_t does not influence the model prices of vanilla European options as they are given by the market implied terminal CDF.
- The kernel process X_t influences the dynamics of the forward volatility skew generated by the model and therefore the prices of exotic options.
- It is also preferable to choose an analytical trackable process X_t to reduce the computational efforts.

The Mapping Function $g(t, x)$

- Define a set of calibration maturities $T_i, i = 1, \ldots, m$
- Define the set kernel collocation points $x_{i,j} = x_j(T_i), j = 1, \ldots, n$ having n interpolation points per maturity T_i.
- Define the spot colloation points by $s_{i,j} = g(t_i, x_{i,j})$
- Calibration: The CDF of X_t should match the market implied CDF of S_t for all collocation points.

$$
\begin{aligned}
F_{X_{T_i}}\left(x_{i,j}\right) &= F_{S_{T_i}}\left(s_{i,j}\right) = F_{S_{T_i}}\left(g(T_i, x_{i,j})\right) \\
\Rightarrow g(T_i, x_{i,j}) &= F^{-1}_{S_{T_i}}\left(F_{X_{T_i}}(x_{i,j})\right)
\end{aligned}
$$

- $F_{X_{T_i}}$ is analytically known or easy to compute.
- $F^{-1}_{S_{T_i}}(x)$ can be calculated from $F_{S_{T_i}}(s)$ via Brent solver.

The Mapping Function $g(t, x)$

The values of the two dimensional mapping function $g(t, x)$ are known for T_i and collocation points $x_{i,j}$ with $s_{i,j} = g(t_i, x_{i,j})$.

Interpolation scheme:

- Linear interpolation in t:

$$s_j(t) = s_{i,j} + \left(s_{i+1,j} - s_{i,j}\right) \frac{t - T_i}{T_{i+1} - T_i}, \forall t \in [T_i, T_{i+1}]$$

- Lagrange interpolation in x:

$$g(t, x(t)) = \sum_{j=1}^{N} s_j(t) l_j(x(t)),\ l_j(x(t)) = \prod_{k=1, j\neq k}^{N} \frac{x(t) - x_j(t)}{x_k(t) - x_j(t)}$$

The Collocation Points

Aim: Stable interpolation under the probability distribution $X(t)$

$$\mathbb{E}\left[S(T_i)\right] = \mathbb{E}\left[g(T_i, X(T_i))\right] + \epsilon$$

Optimal collocation points are the Gauss quadrature abscissæ of the underlying probability distribution of the kernel process $X(t)$.

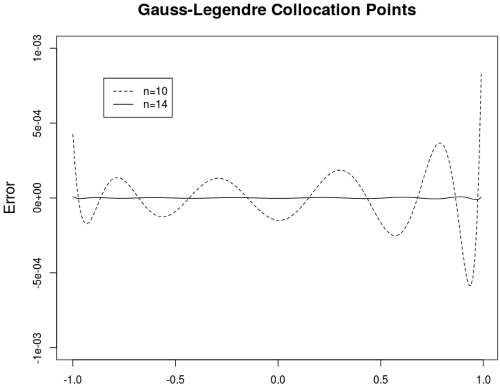
Use the barycentric formulation of the Lagrange interpolation.

The Collocation Points

Example:

- Lagrange interpolation of $f(x) = \frac{e^x}{\cos(x)}$ in $x \in \{-1, 1\}$
- Uniform probability distribution $\Rightarrow$ Gauss-Legendre abscissæ

```
Array x = GaussLegendreIntegration(n).x();
```



The Normal-CLV Model

Choose Ornstein-Uhlenbeck Kernel Process

$$X_t = \kappa(\theta - X_t)dt + \sigma dW_t$$

The exact solution is a normal distribution with

$$\mathbb{E}\left[X(t)\right] = X(t_0)e^{-\kappa t} + \theta\left(1 - e^{-\kappa t}\right), \mathbb{V}ar\left[X(t)\right] = \frac{\sigma^2}{2\kappa}\left(1 - e^{-2\kappa t}\right)$$

Collocation points are therefore given by

$$x_j(t) = \mathbb{E}\left[X_t\right] + \sqrt{\mathbb{V}ar\left[X_t\right]}x_j^{\mathcal{N}(0,1)}$$

with $x_j^{\mathcal{N}(0,1)} = \mathbf{x}$ given by the Gauss-Hermite abscissæ

```
Array x=std::sqrt(2)*GaussHermiteIntegration(n).x();
```

Pricing under the Normal-CLV Model

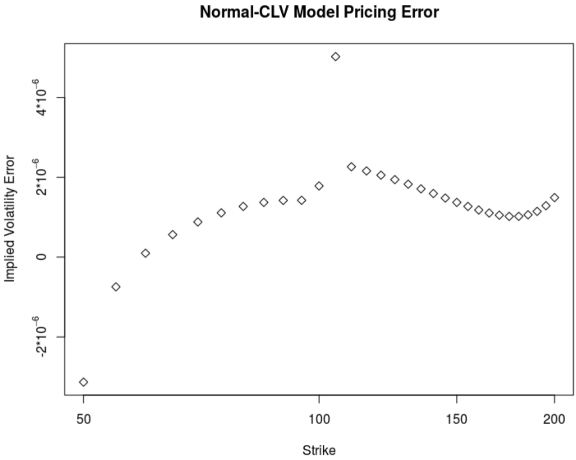
Once the mapping function $g(t, x)$ has been calibrated to the market implied probability distribution pricing is straight forward.

Partial Differential Equation (Feynman-Kac):

$$
\begin{aligned}
\frac{\partial V}{\partial t} + \kappa\left(\theta - x\right)\frac{\partial V}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 V}{\partial x^2} - rV &= 0 \\
V(T, S) &= \text{Payoff}\left(g(T, X(T)\right)
\end{aligned}
$$

Normal-CLV: Pricing Error for Vanilla Options

- Market prices model: $S_0 = 100, r = 0.1, q = 0.04, \sigma = 0.25, T = 1$
- Normal-CLV process: $\kappa = 1.0, \theta = 0.1, \sigma = 0.5, x_0 = 0.1, n = 10$



Normal-CLV: Forward Volatility Skew

Model Setup:

- Market prices are given by Heston model with

$$S_0 = 100, r = 0.1, q = 0.05,$$
$$\nu_o = 0.09, \kappa = 1.0, \theta = 0.06, \sigma = 0.4, \rho = -0.75$$

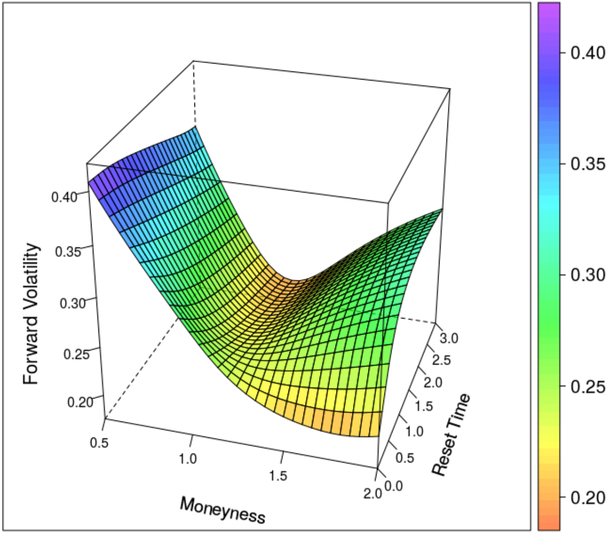
- Normal-CLV process parameters are given by

$$\kappa = 1.0, \theta = 0.1, \sigma = 0.5, x_0 = 0.1, n = 10$$

- Implied volatility of an forward starting European option with maturity date T_2 six month after the reset date T_1

$$\text{Payoff} = \max(0, S_{T_2} - \alpha S_{T_1})$$

Normal-CLV: Forward Volatility Skew



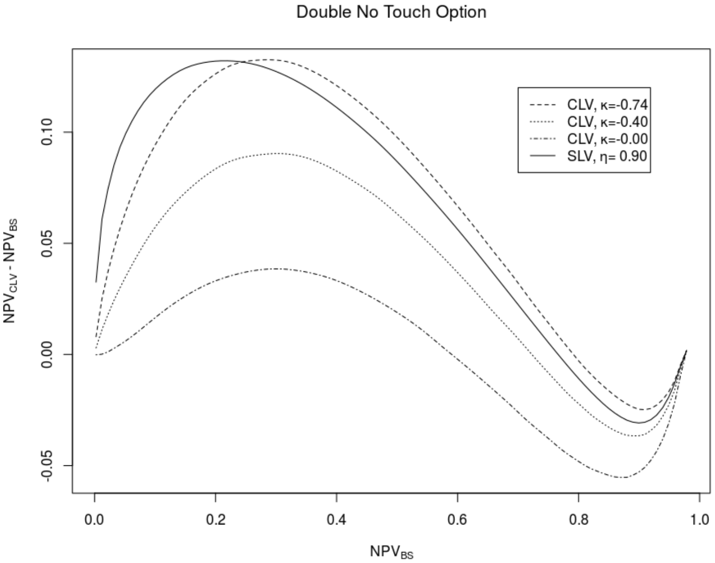
Normal-CLV: Comparison with Heston-SLV

Add leverage function $L(S_t, t)$ and mixing factor η to the Heston Model:

$$\begin{aligned}
d\ln S_t &= \left(r_t - q_t - \frac{1}{2}L(S_t,t)^2\nu_t\right)dt + L(S_t,t)\sqrt{\nu_t}\,dW_t^S \\
d\nu_t &= \kappa\left(\theta - \nu_t\right)dt + \eta\sigma\sqrt{\nu_t}\,dW_t^\nu \\
\rho dt &= dW_t^\nu dW_t^S
\end{aligned}$$

Mixing factor η tunes between stochastic and local volatility.

Normal-CLV: Comparison with Heston-SLV



Square-Root CLV Model

Choose Square-Root kernel process

$$d\nu_t = \kappa\left(\theta - \nu_t\right)dt + \sigma\sqrt{\nu_t}dW$$

The probability density function of ν_t given ν_0 is

$$\nu_t = \frac{\sigma^2(1-e^{-\kappa t})}{4\kappa}\chi_d'^2\left(\frac{4\kappa e^{-\kappa t}}{\sigma^2(1-e^{-\kappa t})}\nu_0\right)$$

where $\chi_d'^2$ denotes the noncentral chi-square random variable with

$$d = \frac{4\theta\kappa}{\sigma^2}$$

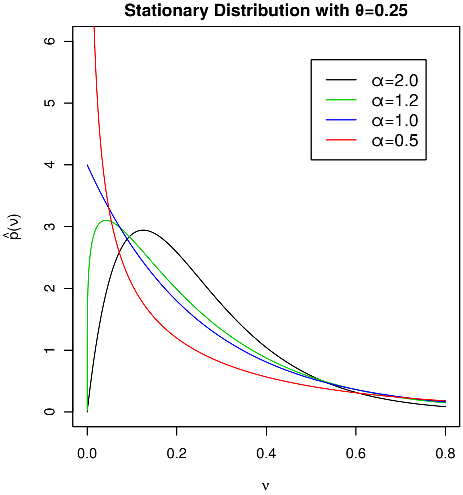
degrees of freedom and the noncentrality parameter

$$\lambda = \frac{4\kappa e^{-\kappa t}}{\sigma^2(1-e^{-\kappa t})}\nu_0$$

Square-Root CLV: Stationary Kernel Distribution

$$\lim_{\nu\to 0}\hat{p}(\nu) = \begin{cases} \infty & \text{if } \alpha < 1 \\ \theta^{-1} & \text{if } \alpha = 1 \\ 0 & \text{if } \alpha > 1 \end{cases}$$

$$\text{with } \alpha = \frac{2\kappa\theta}{\sigma^2}$$



Square-Root CLV: Gaussian Quadrature

The collocation points are given by the eigenvalues of a symmetric, tridiagonal matrix with the diagonal $\{\alpha_i\}$ and the minor diagonal $\{\sqrt{\beta_i}\}$.

$$
\begin{aligned}
z_{k,i} &= z_{k-1,i+1} + \alpha_k z_{k-1,i} - \beta_k z_{k-2,i} \\
\alpha_{k+1} &= \frac{z_{k-1,k}}{k-1, k-1} - \frac{z_{k,k+1}}{z_{k,k}} \\
\beta_{k+1} &= \frac{z_{k,k}}{z_{k-1,k-1}} \\
z_{-1,i} &= 0 \\
z_{0,i} &= \mu_i = \int_0^\infty x^i \chi'^2_{d,\lambda}(x) dx \\
\alpha_1 &= -\frac{\mu_1}{\mu_0} \\
\beta_1 &= \mu_0 = 1
\end{aligned}
$$

Square-Root CLV: Gaussian Quadrature

Mathematica can calculate the first *n* moments

$$\mu_i(d, \lambda) = \int_0^\infty x^i \chi'^2_{d,\lambda}(x) dx$$

and export it as plain C code

```
m[n_] := CForm[ Expectation[X^n, X \[Distributed]
 NoncentralChiSquareDistribution[d, lambda]] // Simplify]
```

The *i*-th moment $\mu_i(d, \lambda)$ is a polynom of order *i* in *d* and λ .

The recurrence relation becomes unstable for *i* is greater than 14. Use Boost.Multiprecision instead of `QuantLib::Real` to solve the equation.

Square-Root CLV: Calibrate Forward Dynamics

Use the parameters $\{\kappa, \theta, \sigma\}$ of the kernel process to calibrate the forward skew dynamics. The mapping function $g(t, x)$ will ensure the correct pricing of European options. The forward skew dynamics is quantified by implied volatiltiy of a set of forward starting option.

Market prices are given by a Heston model with

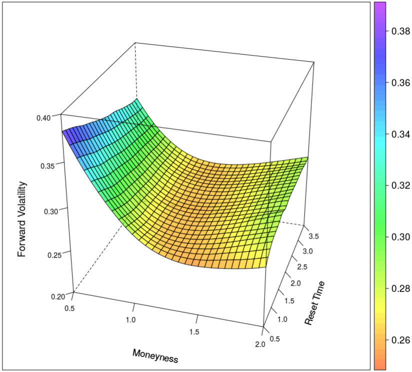
$$S_0 = 100, r = 0.1, q = 0.05,$$
$$\nu_o = 0.09, \kappa = 1.0, \theta = 0.06, \sigma = 0.4, \rho = -0.75$$

Target forward skew dynamics is defined by calibrated Heston-SLV model with mixing angle $\eta = 0.25$ and

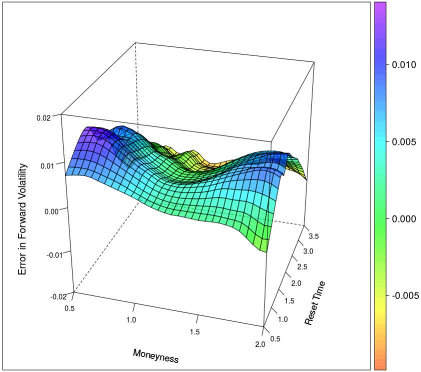
$$\nu_o = 0.09, \kappa = 1.0, \theta = 0.06, \sigma = 0.4, \rho = 0.0$$

Square-Root CLV: Calibrate Forward Dynamics

Implied volatility of an forward starting European option with maturity date six month after the reset date.



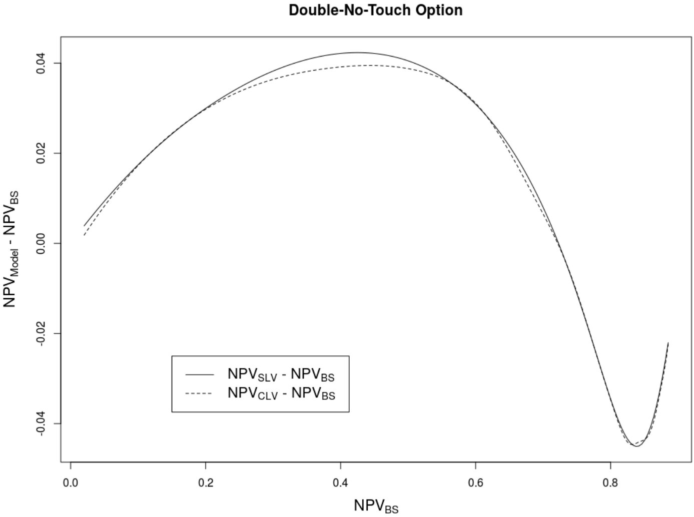
Heston-SLV Forward Volatility



Error CLV Forward Volatility

Square-Root CLV: Double-No-Touch Options

CLV forward volatility dynamics calibrated to Heston SLV dynamics.



Conclusion

- The CLV model allows for fast and accurate calibration to European option prices and efficient pricing of exotic options.
- Only the mean reversion speed influences the forward skew dynamics in the Normal-CLV model.
- The Square-Root kernel process offers more control over the forward skew dynamics.
- Dynamics can be calibrated to more complex models resulting in faster pricing

Literature

Morandi Cecchi and Redivo Zaglia.
Computing the coefficients of a recurrence formula for numerical integration by moments and modified moments.
Journal of Computational and Applied Mathematics, 49(1):207–216, 1993.

Lech A. Grzelak.
The CLV Framework - A Fresh Look at Efficient Pricing with Smile.
2016.
`https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2747541`.

M.Suaŕez-Taboada L.A. Grzelak, J.A.S. Witteveen and C.W. Oosterlee.
The Stochastic Collocation Monte Carlo Sampler: Highly Efficient Sampling from 'Expensive' Distributions.
`https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2529691`.

Disclaimer

The views, opinions, positions or strategies expressed in this presentation are those of the author and do not necessarily represent the views, opinions, positions or strategies of and should not be attributed to Uniper Global Commodities.